BLACK CODHER

CODING PROGRAMME

CODING BLACK FEMALES

Black Codher Bootcamp

NÍ YỌ

BLACK CODHER

CODING PROGRAMME

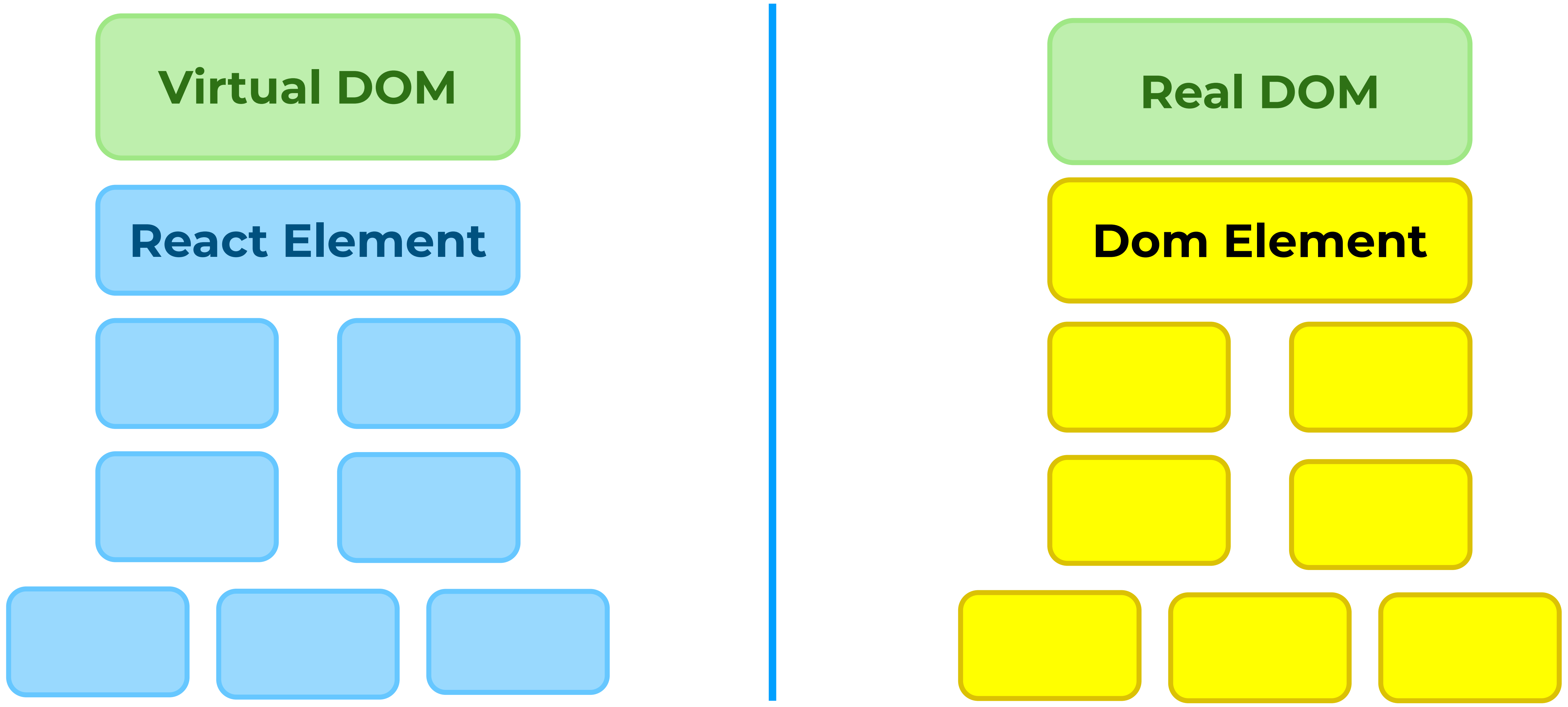
UNIT 4 - Session 2
React

Goals for Unit 4 - Session 2

1. Understand what React components are and how React uses a Virtual DOM
2. Understand what package managers are and the different types
3. Download and install Node.js
4. Create your first React project using the **create-react-app** command
5. Understand what files make up a React web application
6. Understand and run a unit test using the Jest test runner

What is a Component?

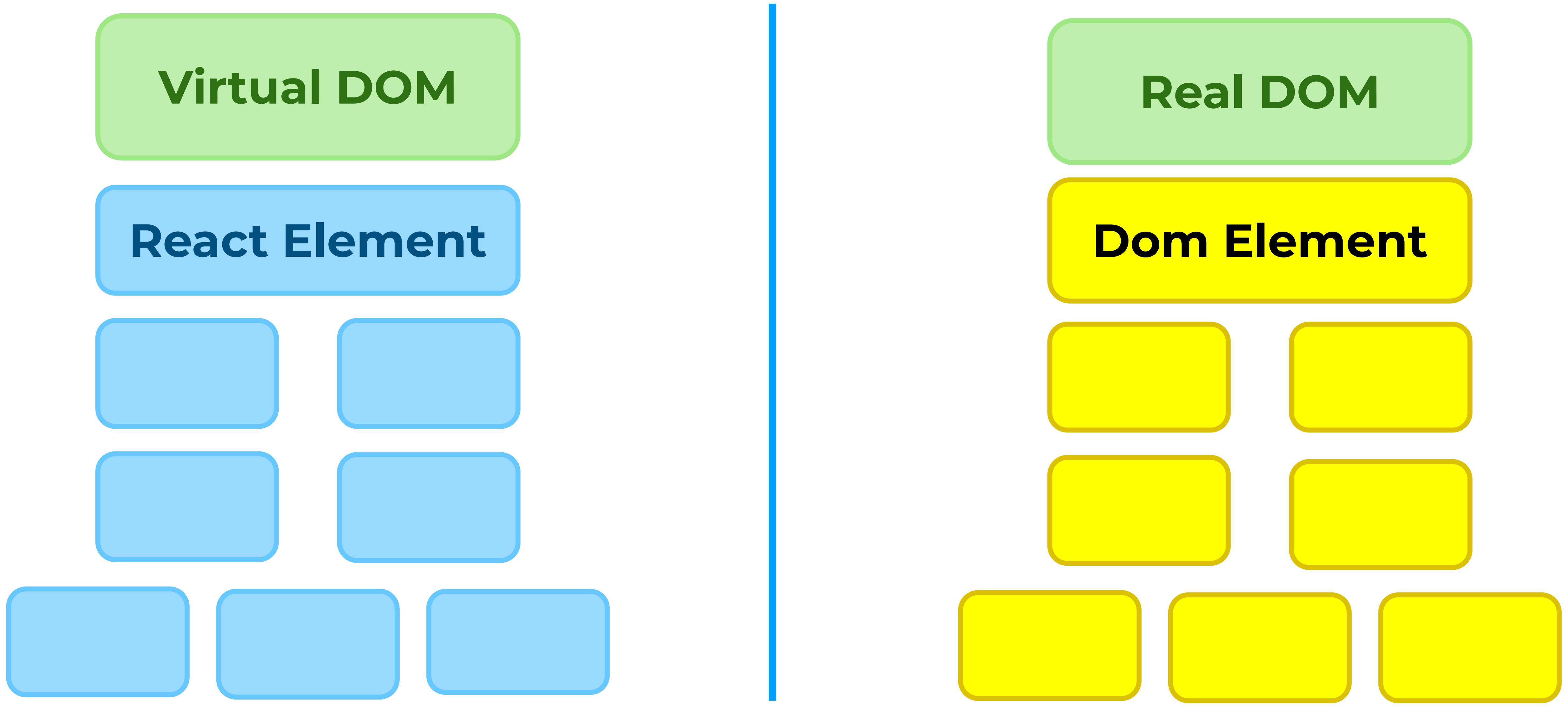
- Components are at the heart of React applications
- React components are added in a tree structure to a Virtual Document Object Model (ReactDOM), which maps to a real DOM element
- With React you build several independent, isolated and reusable components that you can use to construct a user interface



What is a Component?

- JavaScript HTML Document Object Model (DOM) defines a standard on how to select, update, change and delete HTML elements from a page/document.

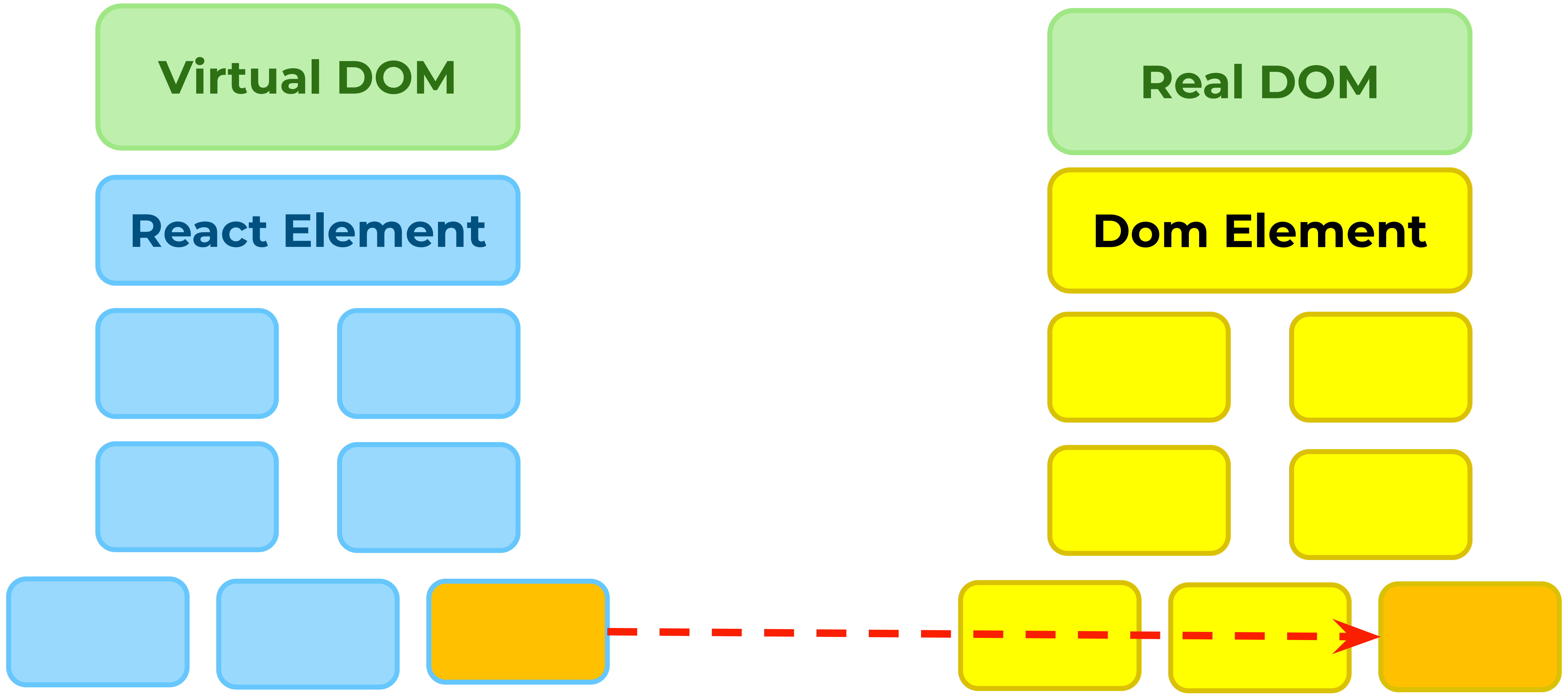
- E.g. document.getElementById() – Standard method to fetch an element by an id from the real DOM.



React Virtual DOM

- In React when a component is changed, a new element is created in the React virtual DOM

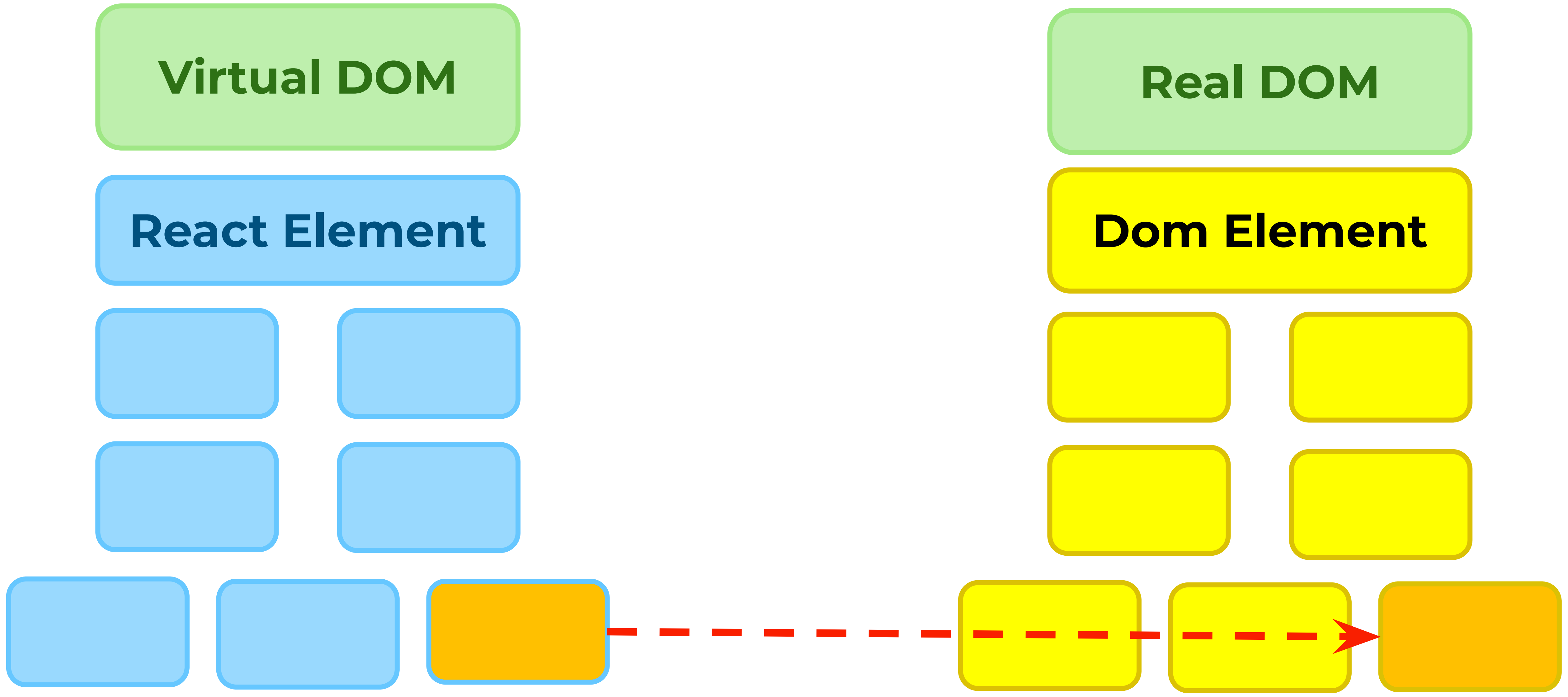
- React keeps track of the element that has changed in the Real DOM then triggers an update to the same element in the Real DOM



React Virtual DOM

- When building applications with React we no longer have to change elements in the Real DOM by attaching handlers.

- When a state changes in React, React will 'react' by changing a component in the Real DOM.



Checkpoint!

How are you feeling?

RED - I have no idea what you’re talking about.

YELLOW - I have some questions but feel like I understand some things.

GREEN - I feel comfortable with everything you’ve said.

Components: LinkedIn

App

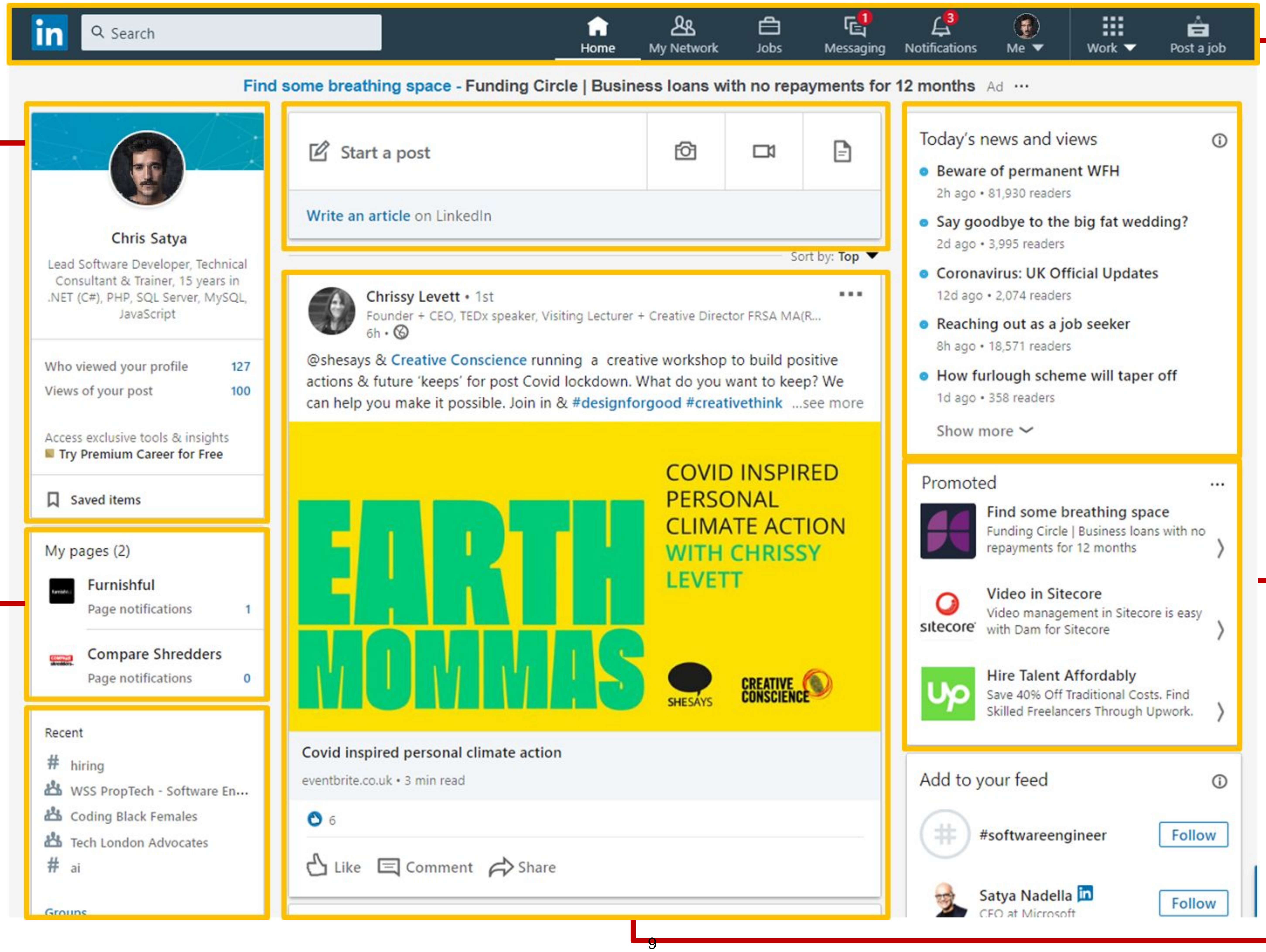
Header

Profile Summary

News Feed

Company Pages

Adverts



Find some breathing space - Funding Circle | Business loans with no repayments for 12 months Ad ···

Chris Satya

Lead Software Developer, Technical Consultant & Trainer, 15 years in .NET (C#), PHP, SQL Server, MySQL, JavaScript

Who viewed your profile	127
Views of your post	100

Access exclusive tools & insights
Try Premium Career for Free

Saved items

My pages (2)

Furnishful Page notifications	1
Compare Shredders Page notifications	0

Recent
- hiring
- WSS PropTech - Software En...
- Coding Black Females
- Tech London Advocates
- ai

Groups

Start a post

Write an article on LinkedIn

Sort by: Top

Chrissy Levett • 1st
Founder + CEO, TEDx speaker, Visiting Lecturer + Creative Director FRSA MA(R...
6h •

@shesays & Creative Conscience running a creative workshop to build positive actions & future 'keeps' for post Covid lockdown. What do you want to keep? We can help you make it possible. Join in & #designforgood #creativethink ...see more

Covid inspired personal climate action
eventbrite.co.uk • 3 min read

6

Like Comment Share

Today's news and views
- Beware of permanent WFH
  2h ago • 81,930 readers
- Say goodbye to the big fat wedding?
  2d ago • 3,995 readers
- Coronavirus: UK Official Updates
  12d ago • 2,074 readers
- Reaching out as a job seeker
  8h ago • 18,571 readers
- How furlough scheme will taper off
  1d ago • 358 readers

Show more

Promoted

Find some breathing space
Funding Circle | Business loans with no repayments for 12 months

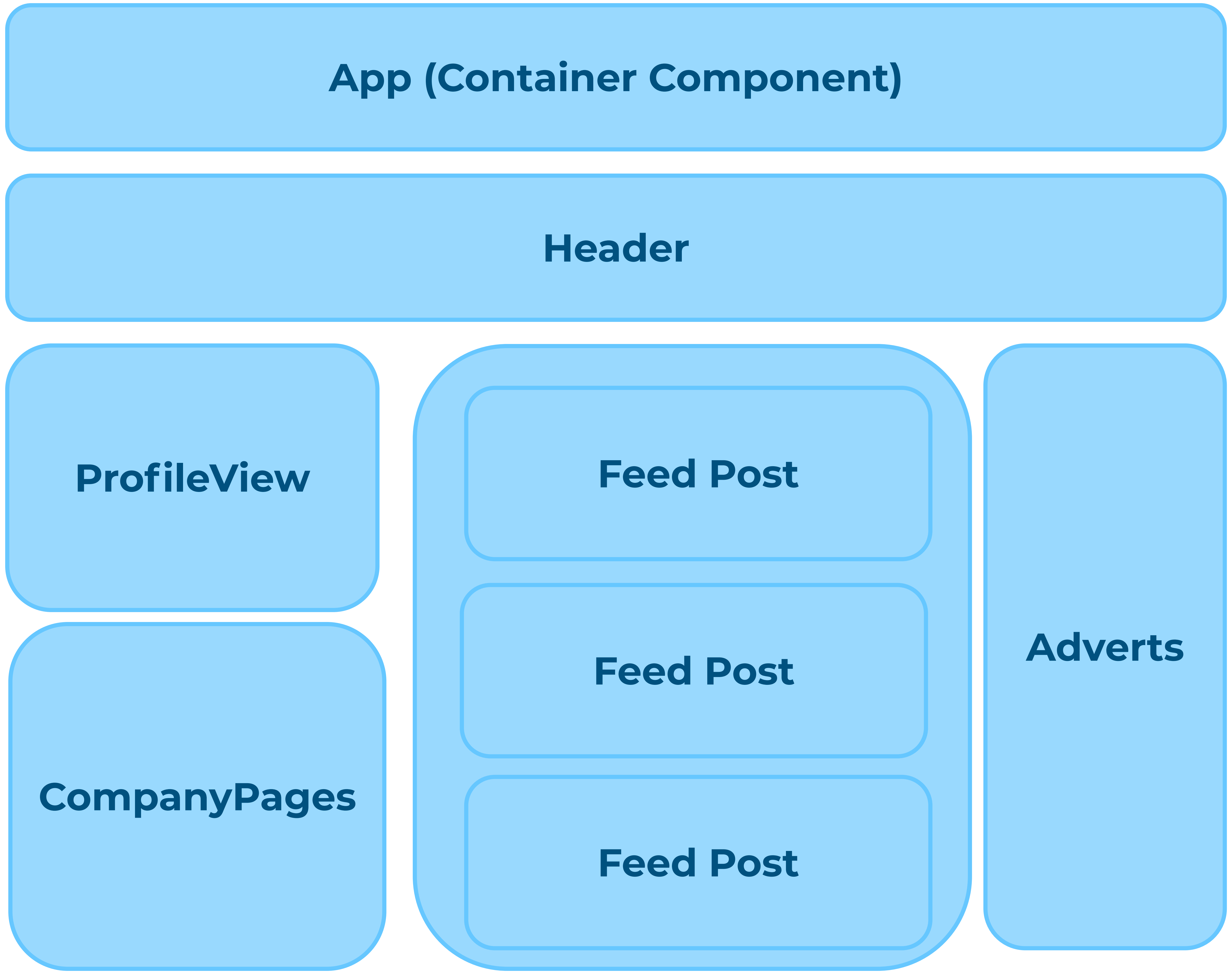
Video in Sitecore
Video management in Sitecore is easy with Dam for Sitecore

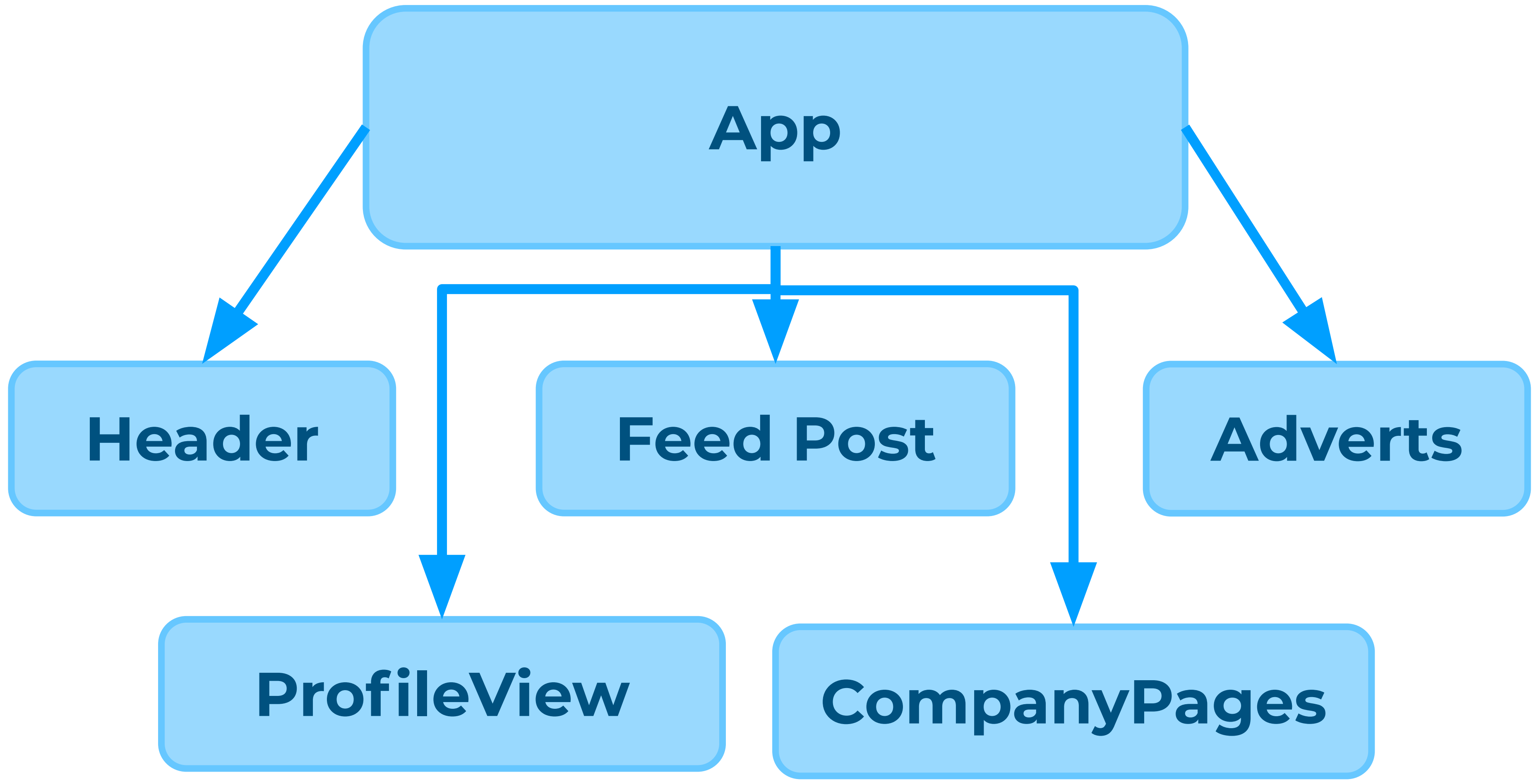
Hire Talent Affordably
Save 40% Off Traditional Costs. Find Skilled Freelancers Through Upwork.

Add to your feed

#softwareengineer	Follow
Satya Nadella CEO at Microsoft	Follow

Component Tree: LinkedIn



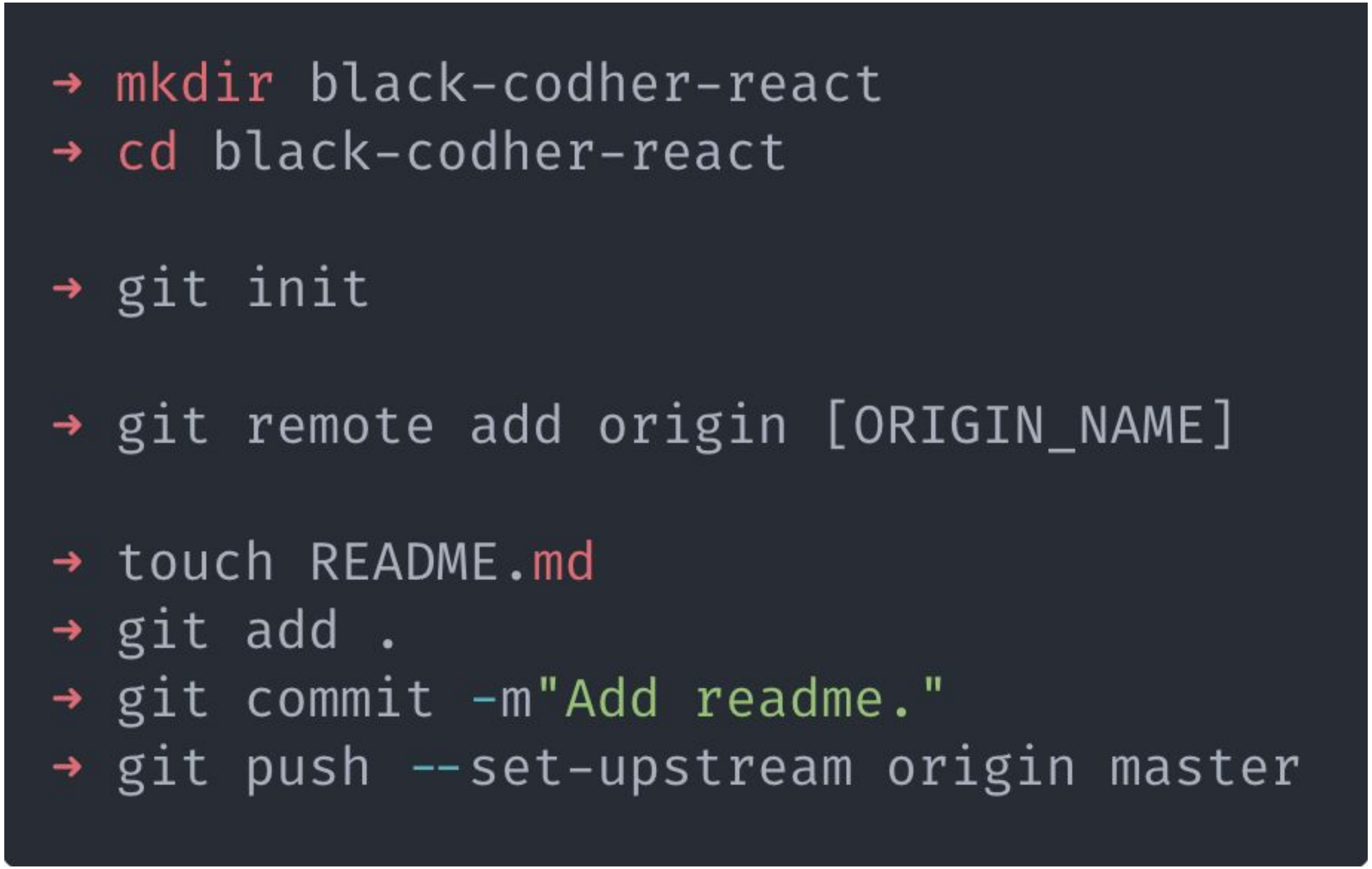


Mockup of Component Markup:

```
<App>
   <Header />
   <ProfileView />
   <Feeds>
      <FeedPost />
      ...
   </Feeds>
   <ProfileView />
   <CompanyPages />
   <Advert />
</App>
```

Create React App Command

Setting up our project folders



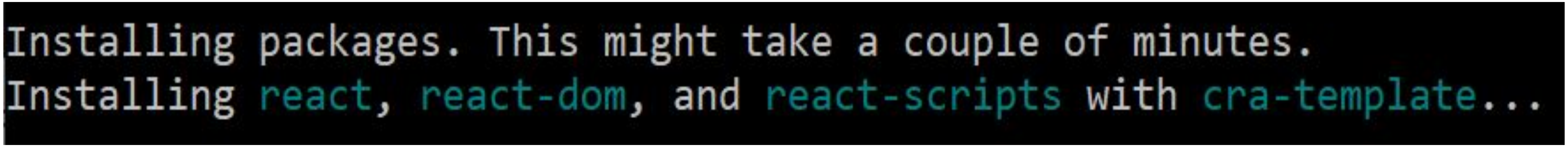
```
→ mkdir black-codher-react
→ cd black-codher-react

→ git init

→ git remote add origin [ORIGIN_NAME]

→ touch README.md
→ git add .
→ git commit -m"Add readme."
→ git push --set-upstream origin master
```

Creating the Application

- **Create-React-App** is a command line tool that sets up a boiler-plate development environment with the latest JavaScript features.

- To create a project using **create-react-app** run the following from a command prompt of VS terminal window (myfirstapp is the name of the folder that will be created):

```
> npx create-react-app myfirstapp
```



```
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
```

Creating the Application

- **create-react-app** creates a frontend build pipeline using **Babel** (babeljs.io) and **Webpack***.

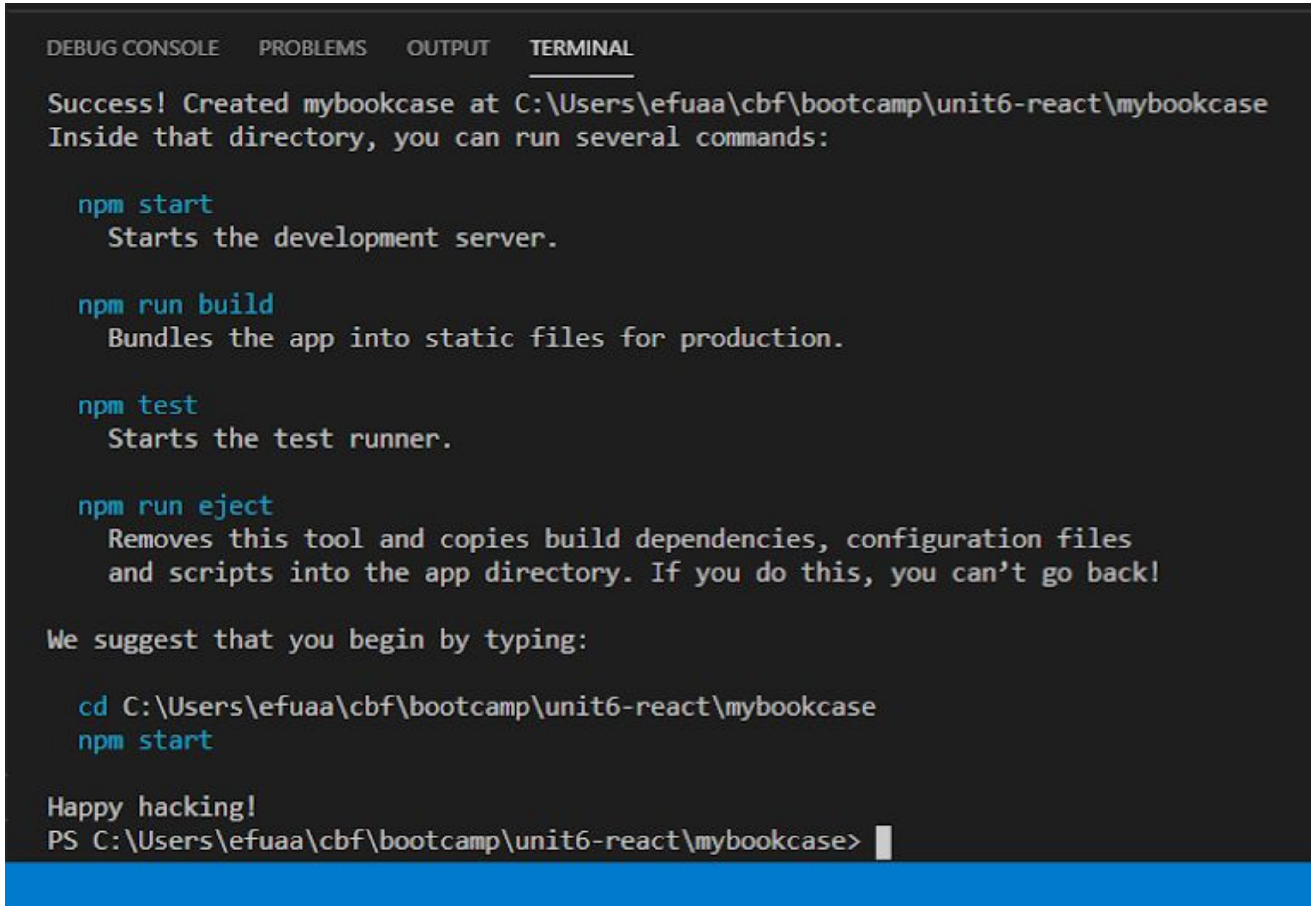
- This will install a zero-config basic application.

- The build tools mean the project will reload whenever you make a change.

- It also already has a server so when you start the app you can test through a browser such as Chrome.

NPM Build and Run Commands

- NPM run, build, test and compile commands:



```
Success! Created mybookcase at C:\Users\efuaa\cbf\bootcamp\unit6-react\mybookcase
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can’t go back!

We suggest that you begin by typing:

  cd C:\Users\efuaa\cbf\bootcamp\unit6-react\mybookcase
  npm start

Happy hacking!
PS C:\Users\efuaa\cbf\bootcamp\unit6-react\mybookcase>
```

Starting the Application

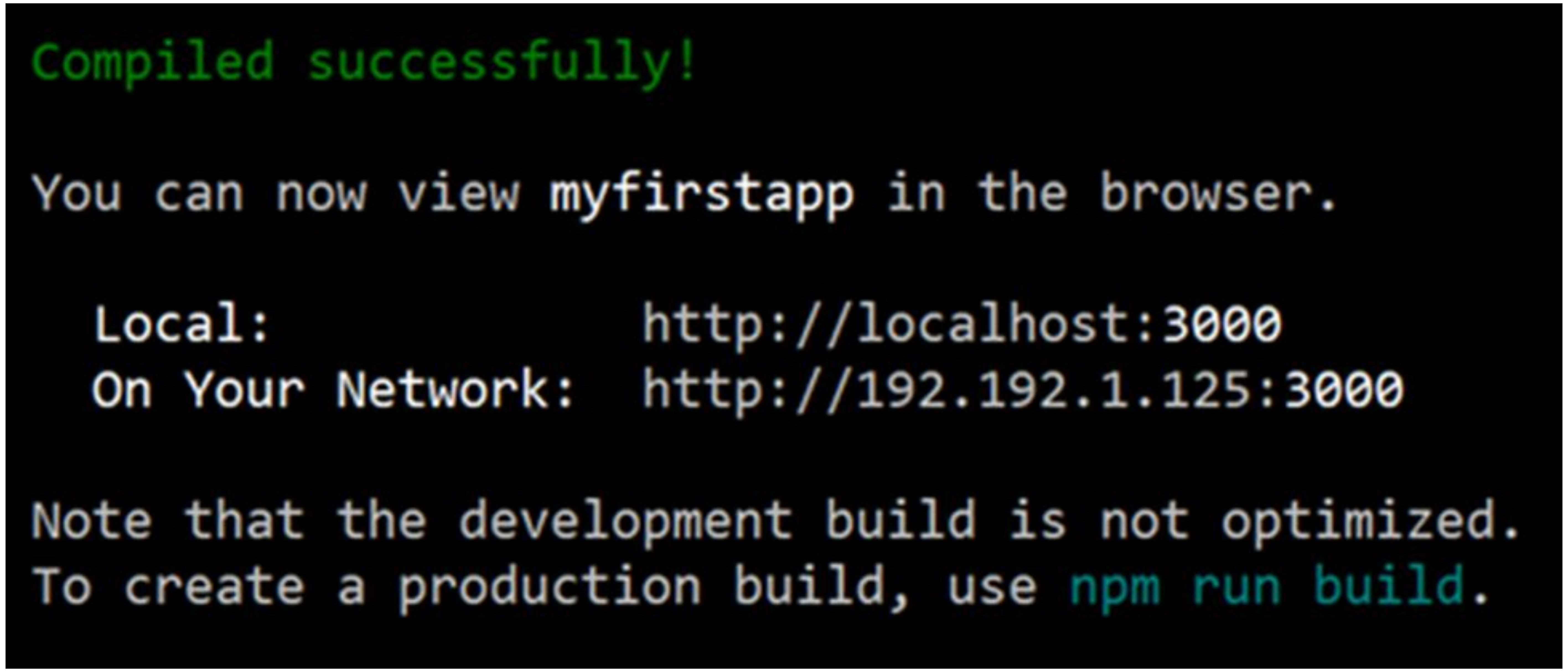
- To start the application, after installation is complete run the following from the command line or the VS Code terminal window:

```
> cd myfirstapp

> npm start
```

- **npm start** starts the application in development mode
- It will run from a local server (http://localhost:3000)

Starting the Application

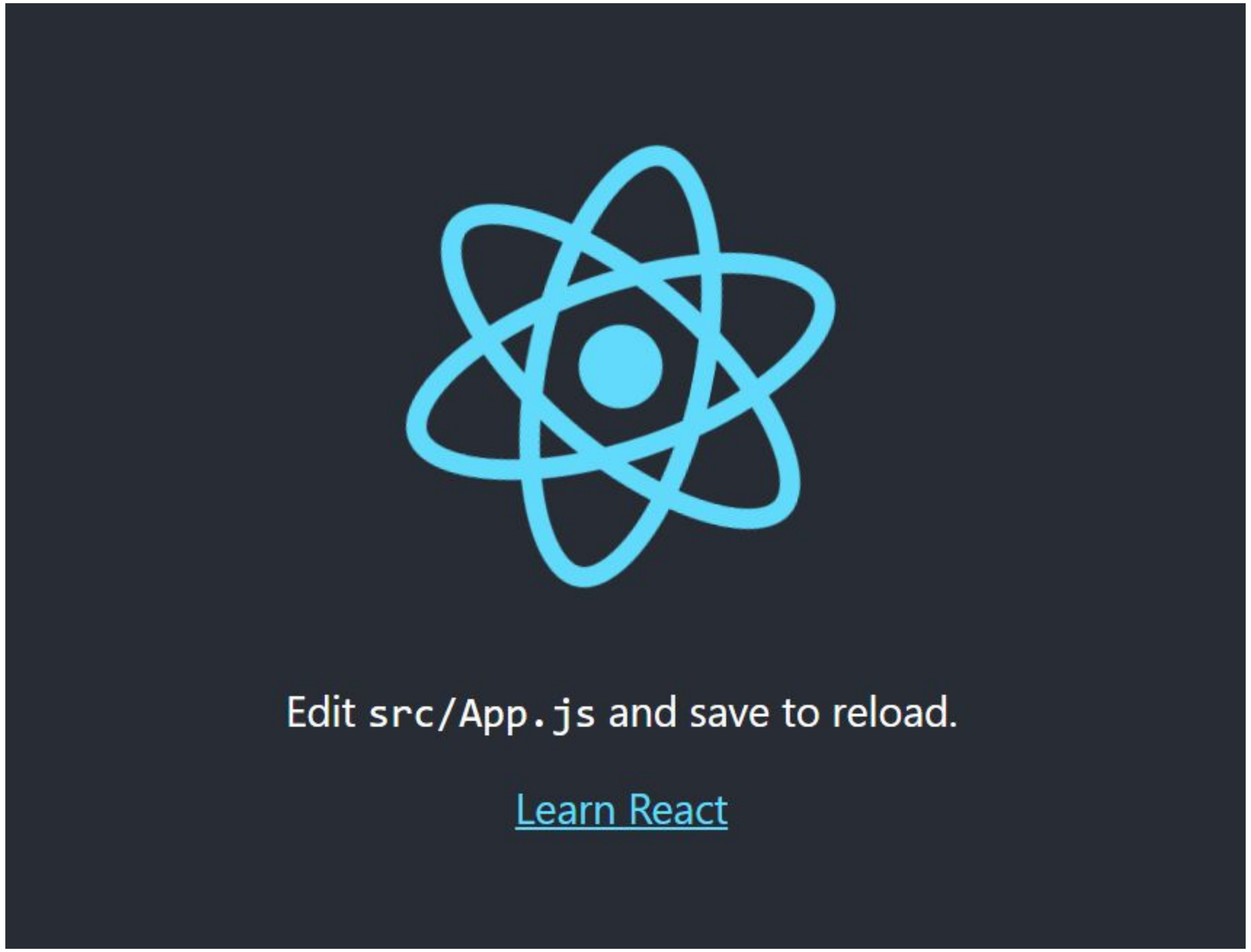
- The other address is your local server on the current network.
- If you open the relevant ports, other people on your shared network will be able to use this address to view your application*.



```
Compiled successfully!

You can now view myfirstapp in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.192.1.125:3000

Note that the development build is not optimized.
To create a production build, use npm run build.
```

Browse App

- The basic web application displayed below:



Edit src/App.js and save to reload.

Learn React

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about.

YELLOW - I have some questions but feel like I understand some things.

GREEN - I feel comfortable with everything you've said.

Understanding Package Mangers

What is NPM?

- Before we create our first full React app using the **create-react-app** helper command, it’s important understand a little about **Node Package Manager (NPM)**

What is NPM?

- **npm** (node package manager) is the dependency/package manager you get out of the box when you install **Node.js**

- It provides a way for developers to install packages and modules both globally and locally

What is NPM?

- Installing a package with **NPM** will install the package dependencies in your local **node_modules** folder

- It’s also a **CLI** (Command Line Interface) that helps you manage the versions of the packages and their dependencies

What is Node.js!?

- Node is an **open source** runtime environment for executing **JavaScript** code outside of a browser window (on a server)

What is NPM?

What does open source mean?

- **Open source**, means software where the original source code is made freely available.

What about NPX? (we used it earlier: `npx create-react-app myfirstapp`)

- NPX is a package runner tool that comes with npm 5.2+

- Installing a package with **NPM** will install the package dependencies in your local node_modules folder.

- **NPX** allows you to run the specific packages that haven't been previously installed.

NPM vs NPX vs YARN

Other package managers exist: YARN

- Yarn is another node package manager used to manage project dependencies. Yarn was created by Facebook.

- When released initially it had better performance than npm, however subsequent versions of npm have bridged the gap in performance.

Exercise 1

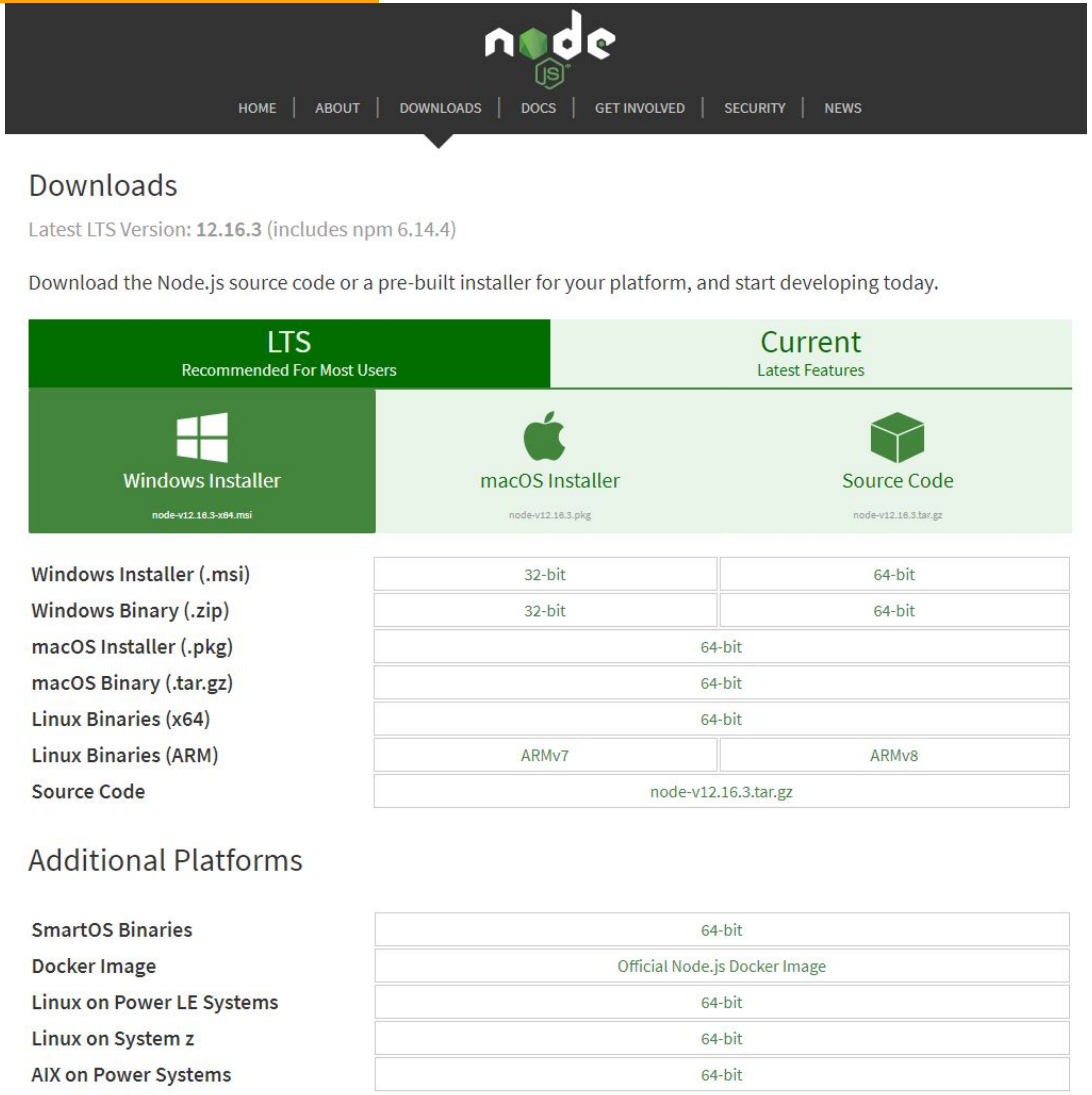
Installing Node.js (NPM by default)

1. Visit: https://nodejs.org/en/download/

2. Download the latest version of Node.js.

   - Windows Installer (.msi)
   - macOS Installer (.pkg)

3. Download and double click to install.



Downloads

Latest LTS Version: **12.16.3** (includes npm 6.14.4)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit	
macOS Binary (.tar.gz)	64-bit	
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v12.16.3.tar.gz	

Additional Platforms

SmartOS Binaries	64-bit
Docker Image	Official Node.js Docker Image
Linux on Power LE Systems	64-bit
Linux on System z	64-bit
AIX on Power Systems	64-bit

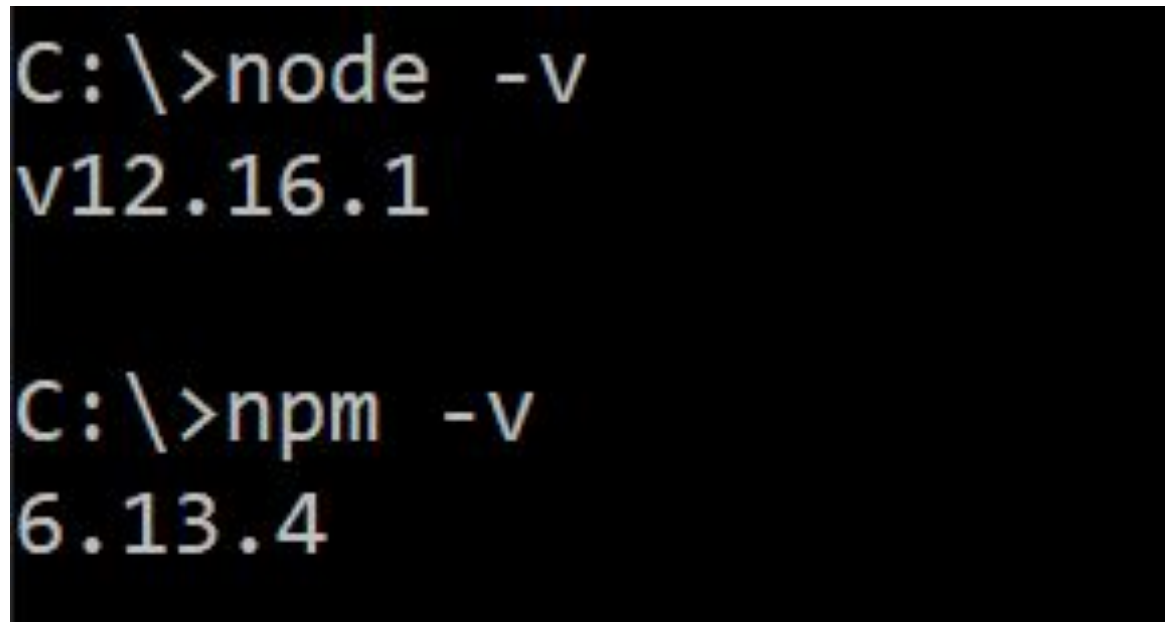
Installing and Running Node.js

- Verify Node.js is installed by checking the version:

```
> node -v
```

- Do the same for NPM:

```
> npm -v
```



```
C:\>node -v
v12.16.1

C:\>npm -v
6.13.4
```

Exercise 2

Exercise 2: Create React App

1. Create your own React App by running the command:

```
> npx create-react-app exercise2
```

2. Open the **index.html** file under the **public** folder and change the **title tag** to "**React Exercise 2**". Refresh your browser window to view changes.

3. Open the **App.css** file under the **src** directory and update the application background colour.

4. Open **App.js** and change the displayed text by removing the **<p>** and **<a>** elements and adding your own **<h1>** element.

Exercise 2: Create React App

BLACK
CODHER

5. (Optional) Create an inline **style** for you new **<h1>** element in **App.js** using curly bracket notation e.g.

Hint: code for inline style:

```
const h1Style = { "fontSize":"30px", "color": "#cc0000",
"textDecoration": "uppercase", "textDecoration":"underline" };
```

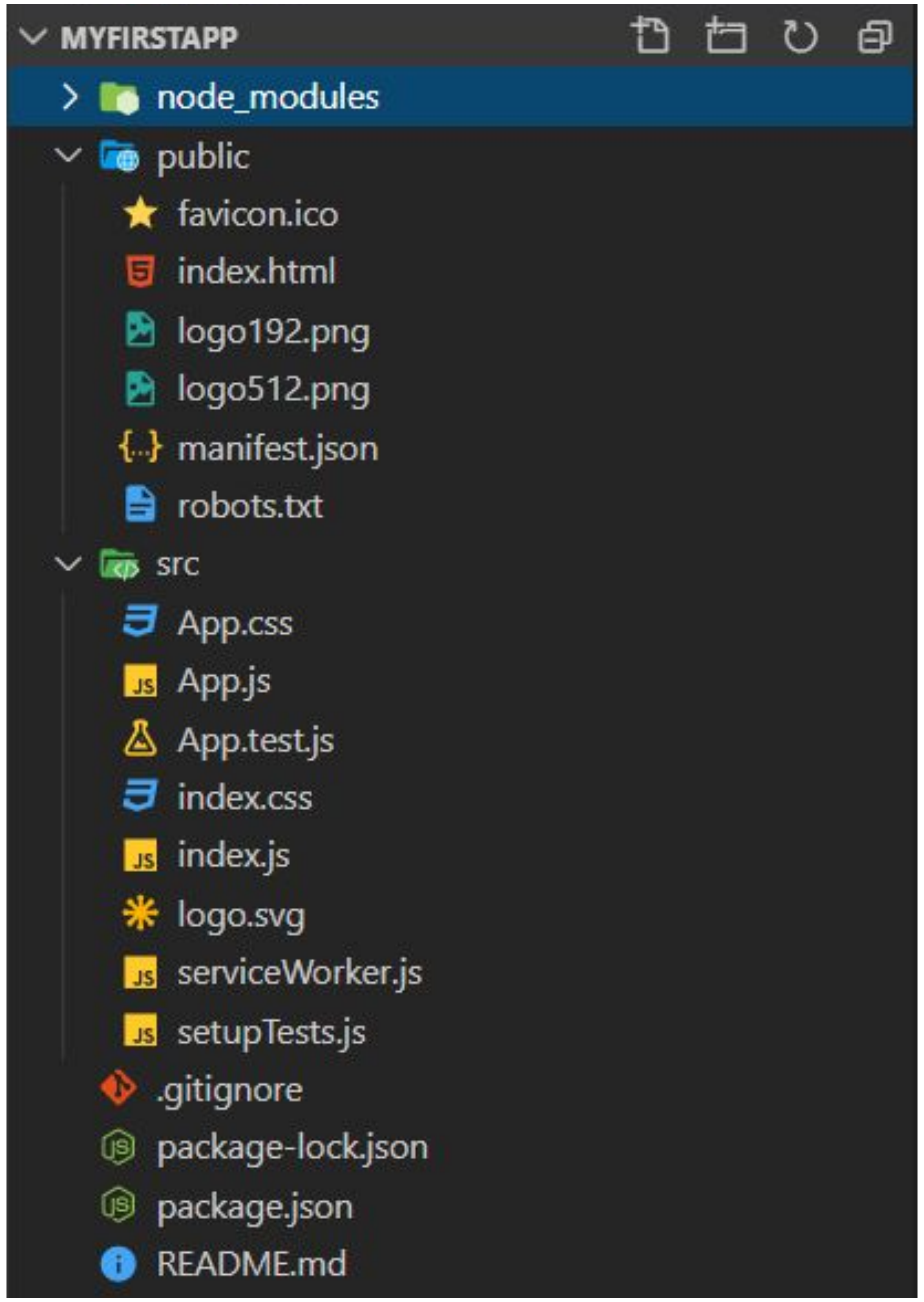
6. (Optional) Set the **h1** element **className**

Anatomy of a React App

Anatomy of a React App

General files and folders:

- Once installed the **create-react-app** contains the following files and folders.
- The **public** folder contains the main HTML and image files for running the site.
- You can alter the **index.html** file to make changes to the loading page for the React app.

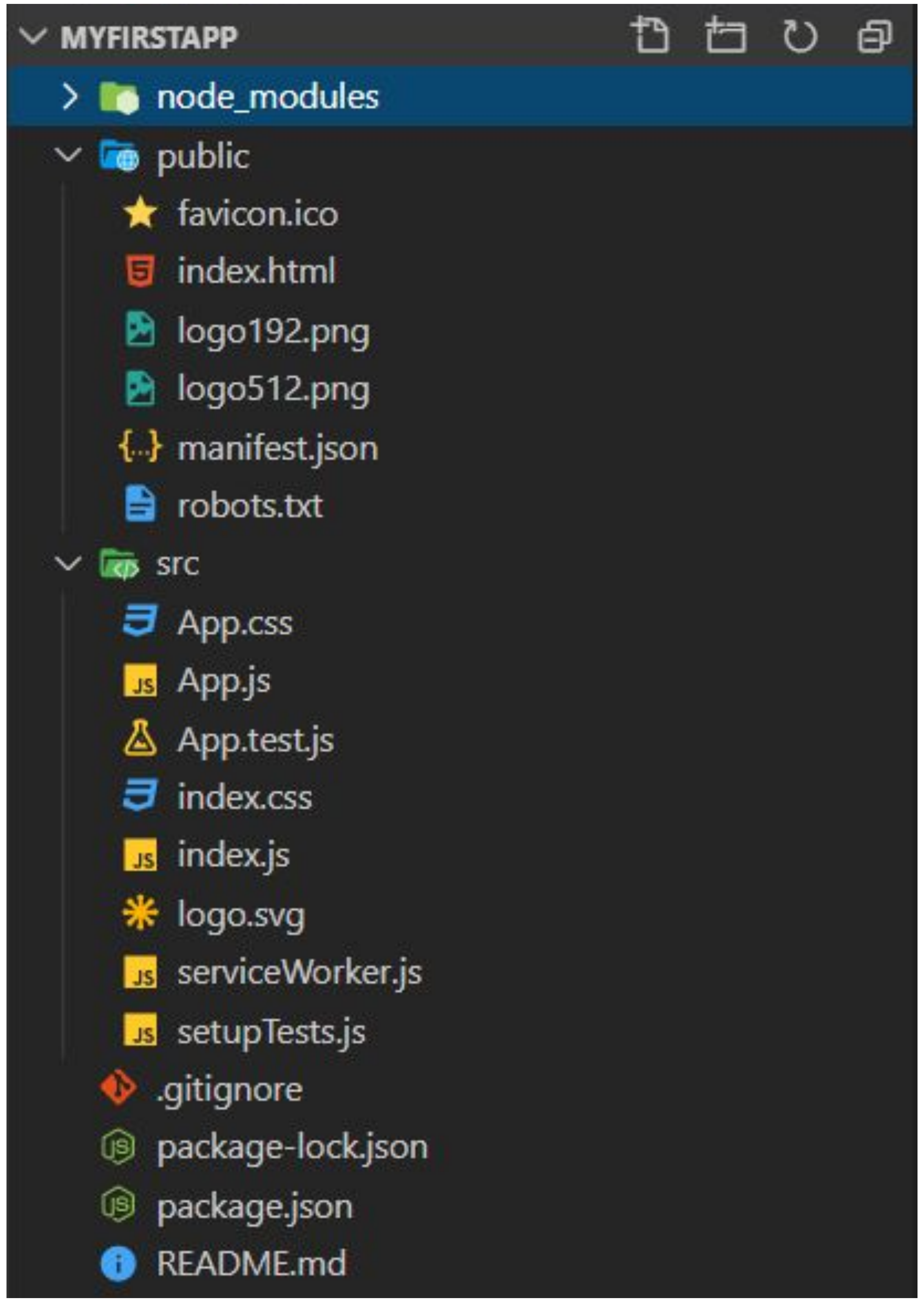


```
MYFIRSTAPP
  > node_modules
  v public
      favicon.ico
      index.html
      logo192.png
      logo512.png
      manifest.json
      robots.txt
  v src
      App.css
      App.js
      App.test.js
      index.css
      index.js
      logo.svg
      serviceWorker.js
      setupTests.js
    .gitignore
    package-lock.json
    package.json
    README.md
```

Anatomy of a React App

General files and folders:

- The **src** folder contains main scripting files to execute the application logic.
- The **src** folder is where you set up conditions, add classes, components and functions to drive the behaviour of your application.
- To change the application look and feel you can change the **App.css** in the **src** folder.

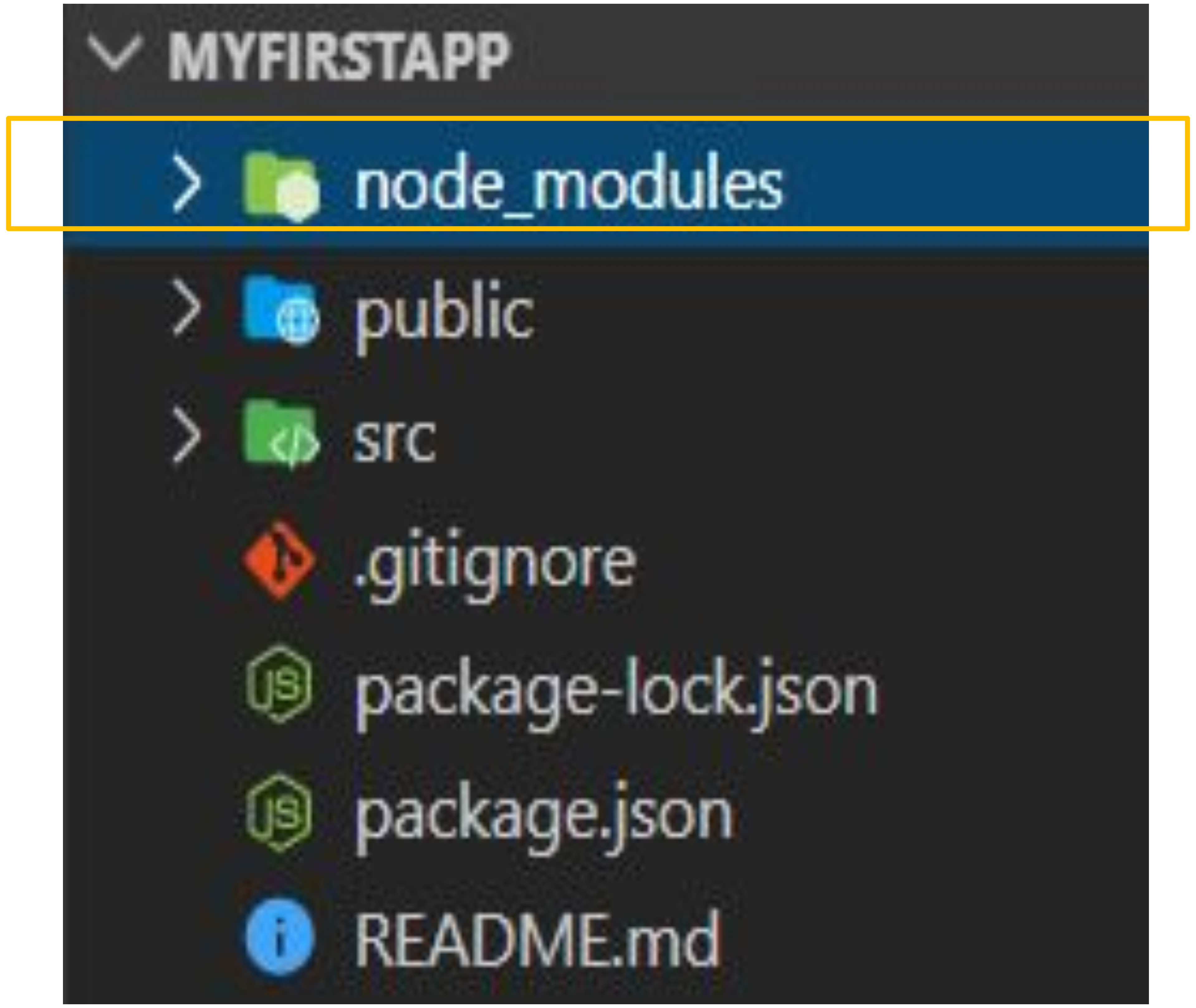


```
MYFIRSTAPP
> node_modules
v public
    favicon.ico
    index.html
    logo192.png
    logo512.png
    manifest.json
    robots.txt
v src
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    serviceWorker.js
    setupTests.js
  .gitignore
  package-lock.json
  package.json
  README.md
```

Anatomy of a React App

node_modules:

- The **node_modules** folder contains all the local packages for React and ReactDom
- Packages installed in this folder can be required using the import call in React.
- It may be an idea to ensure this folder is excluded from git (add to .gitignore)



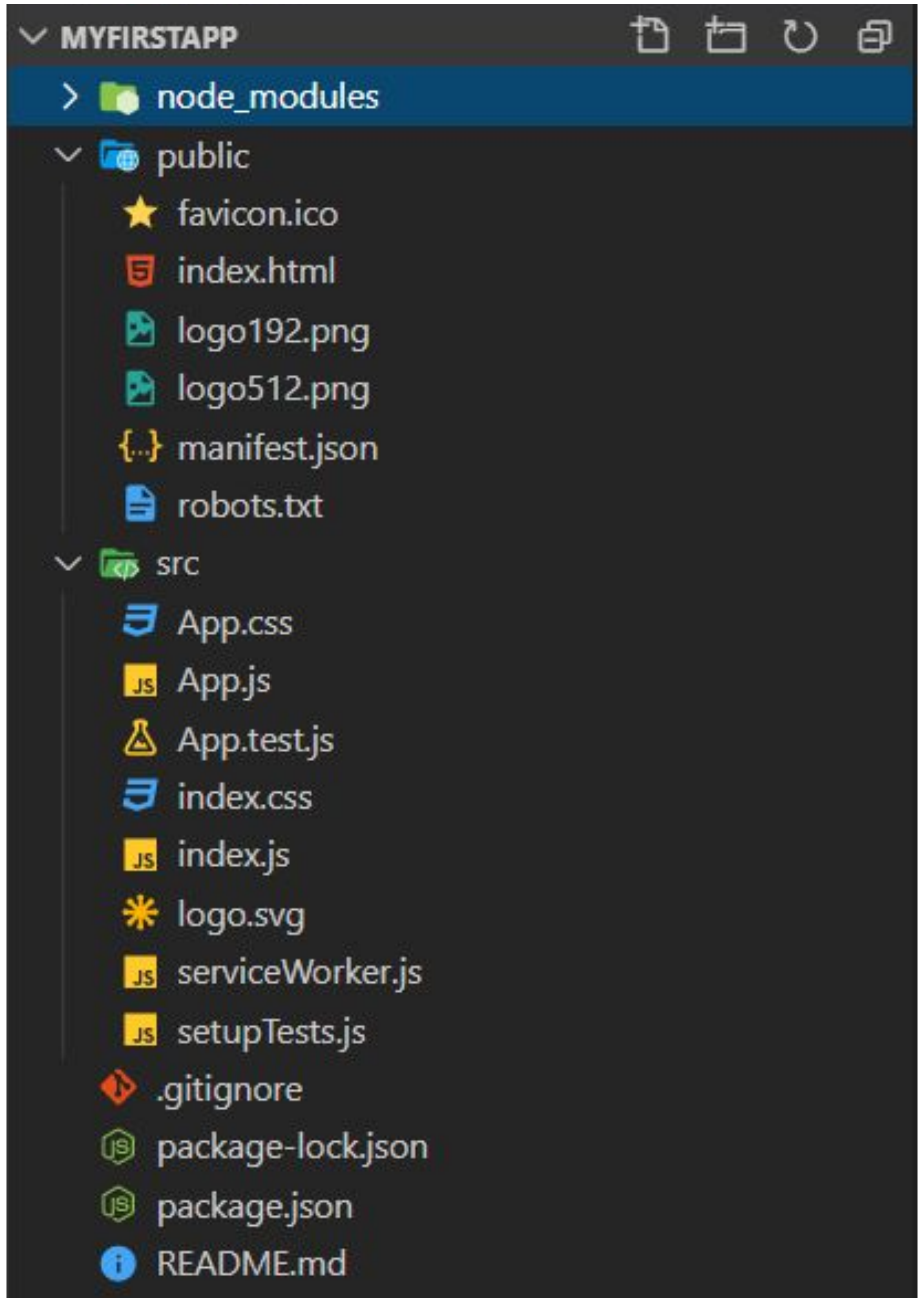
```
MYFIRSTAPP
  > node_modules
  > public
  > src
    .gitignore
    package-lock.json
    package.json
    README.md
```

Create-React-App Files in Brief

Quick look at files
Assuming you are familiar with any *.html, *.js and image (gif, png, svg). Brief explanation of files you may not be familiar with:

robots.txt
Webmasters create this file to instruct robots (search engine robots) how to crawl pages on their website.

favicon.ico
Short icon, website icon is a file containing one or more small icons. Browsers that support favicons display them in the website address bar.



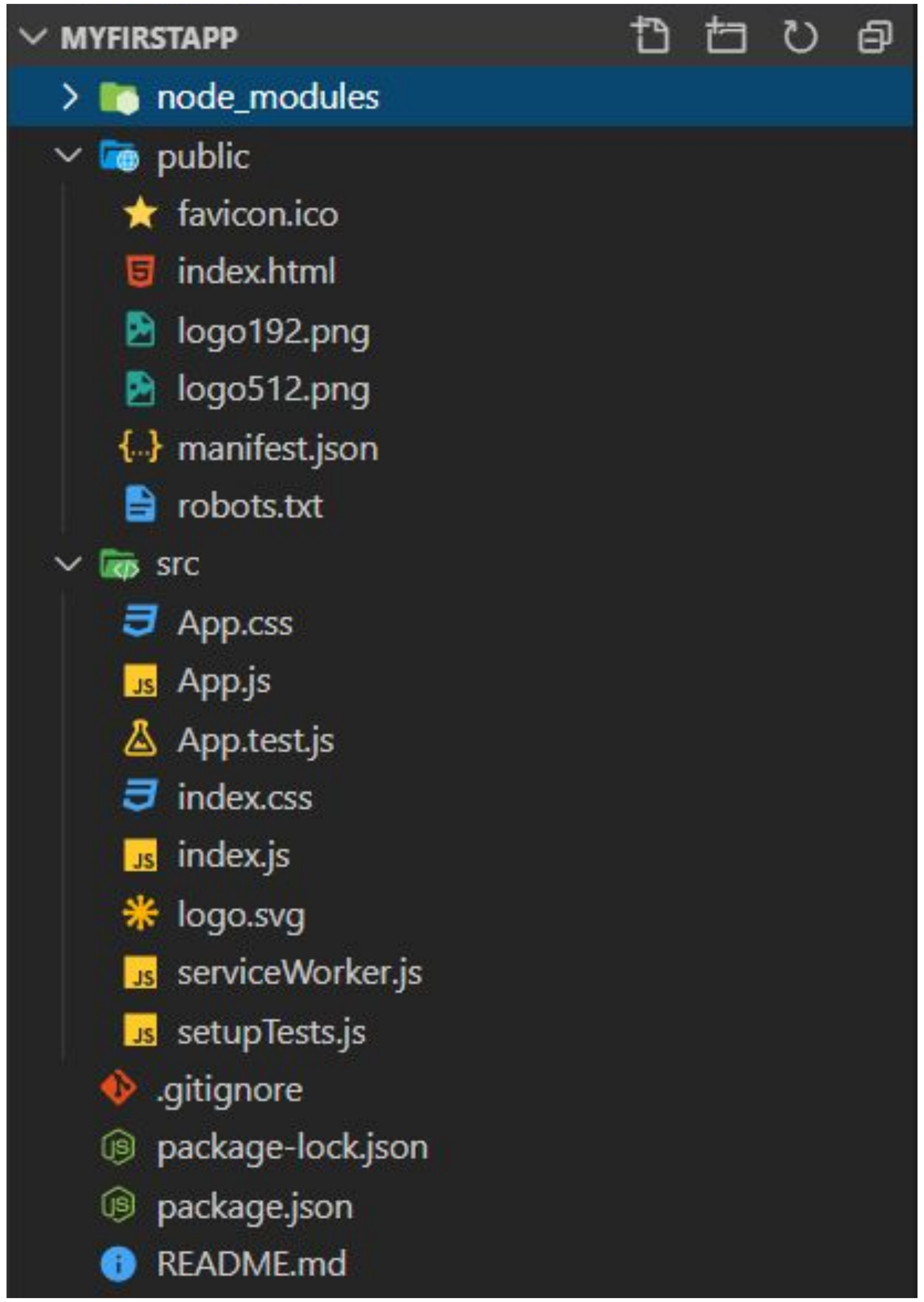
```
MYFIRSTAPP
> node_modules
v public
    favicon.ico
    index.html
    logo192.png
    logo512.png
    manifest.json
    robots.txt
v src
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    serviceWorker.js
    setupTests.js
  .gitignore
  package-lock.json
  package.json
  README.md
```

Create-React-App Files in Brief

Quick look at files

manifest.json
Contains metadata about your extensions such as the name and version. It's a JSON formatted file with comments*

README.md
.md stand for Mark Down. It is a file with simple mark-down language (for bold, italicising) text. It is a form of documentation used to describe other files in a directory or archive.



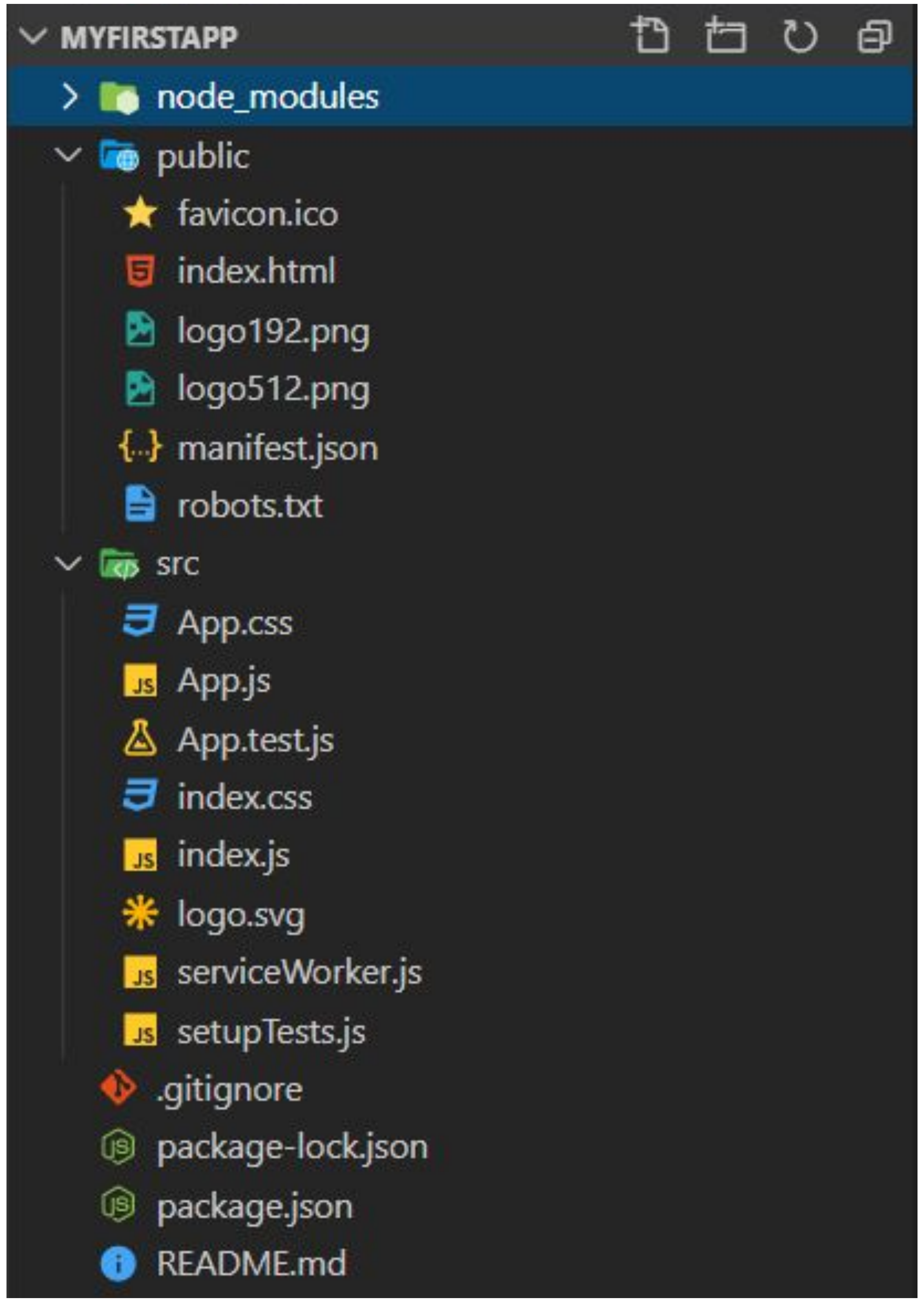
```
MYFIRSTAPP
> node_modules
v public
    favicon.ico
    index.html
    logo192.png
    logo512.png
    manifest.json
    robots.txt
v src
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    serviceWorker.js
    setupTests.js
  .gitignore
  package-lock.json
  package.json
  README.md
```

Create-React-App Files in Brief

Quick look at files

App.test.js

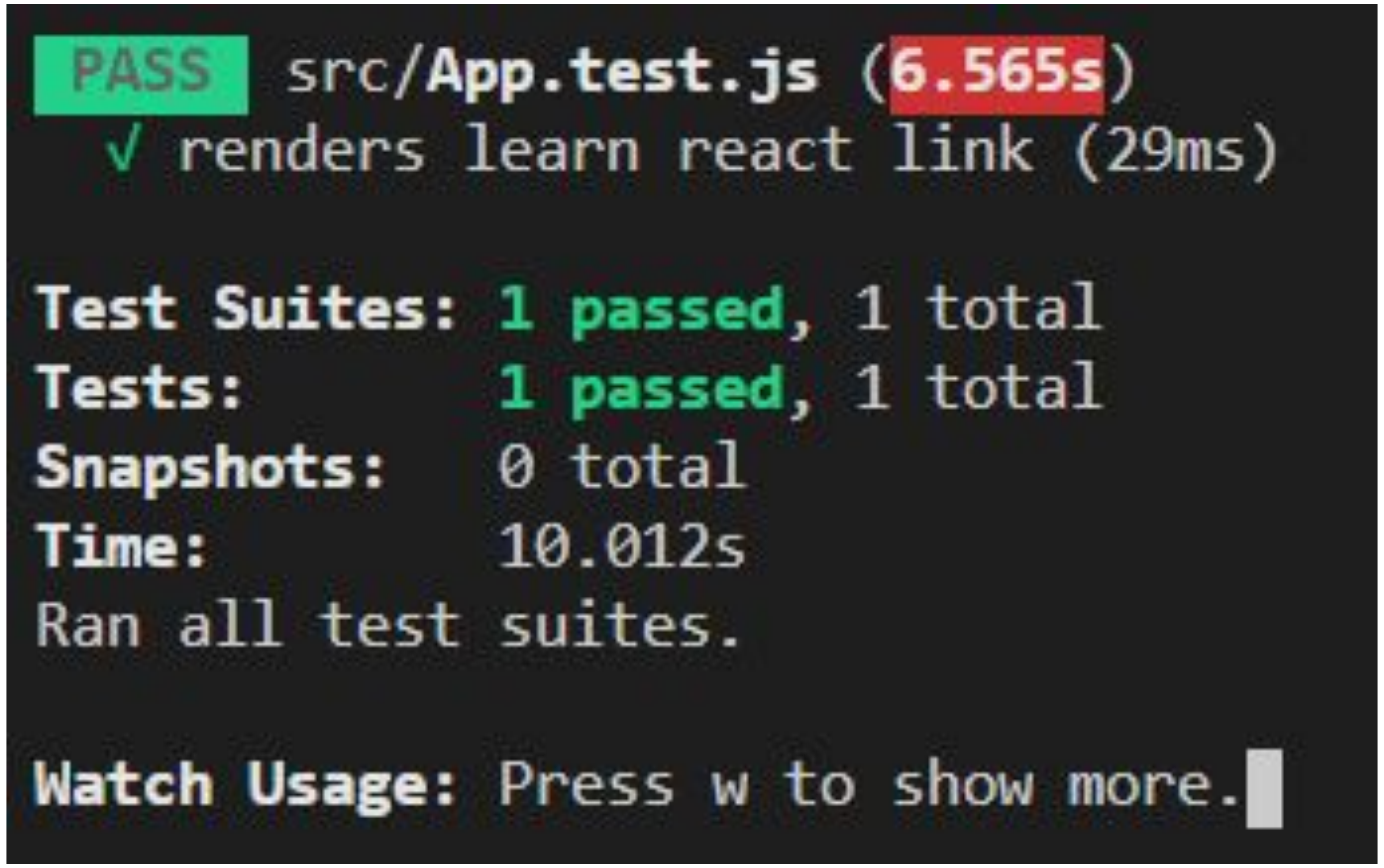
- Create React App uses **Jest** as its test runner
- Any files with **.js** suffix in **__tests__** folders or any files anywhere with a **.test.js** or **.spec.js** suffix will be run as test files by Jest



```
MYFIRSTAPP
> node_modules
v public
    favicon.ico
    index.html
    logo192.png
    logo512.png
    manifest.json
    robots.txt
v src
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    serviceWorker.js
    setupTests.js
  .gitignore
  package-lock.json
  package.json
  README.md
```

A bit about testing...

- To run a test, type **> npm test** in the terminal and follow the instructions

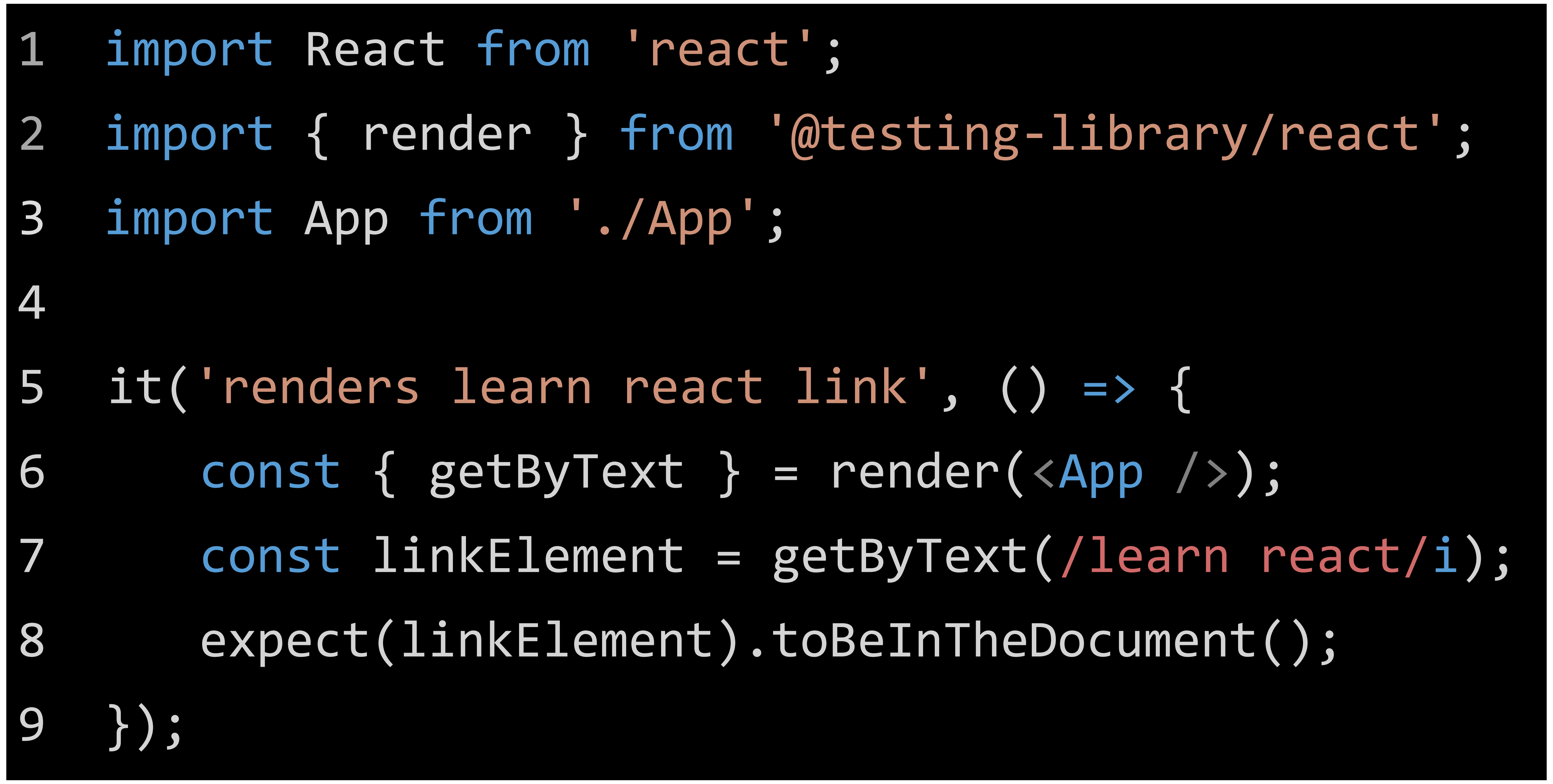


```
 PASS  src/App.test.js (6.565s)
  √ renders learn react link (29ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        10.012s
Ran all test suites.

Watch Usage: Press w to show more.
```

- The test runner shows that the "renders learn react link" test was run
- How long it took to run the test in red (6.565s)
- How many test were run (1 total) and how many tests passed (**1 passed**)
- The test runner also 'watches' the project files and re-runs when changes are made to the project files

A bit about testing...

- The test code can be found in **App.test.js**



```
1  import React from 'react';
2  import { render } from '@testing-library/react';
3  import App from './App';
4
5  it('renders learn react link', () => {
6      const { getByText } = render(<App />);
7      const linkElement = getByText(/learn react/i);
8      expect(linkElement).toBeInTheDocument();
9  });
```

- The test code on line 5 to 9 is looking for a piece of text on the page that reads ‘learn react’
- The syntax **/learn react/i** is special syntax called a regular expression. The ‘**i**’ signifies that the text ‘learn react’ can be upper or lower case
- Line 8 is the test condition which **expect**s the text **.toBeInTheDocument()**
- **it()**, **expect()** and **.toBeInTheDocument()** are methods in the React testing library referenced on line 2

Summary:

1. Understand what React components are and how React uses a virtualDOM instead of the real DOM (JQuery, Vanilla Javascript)

2. Understand what package managers are and the different types (NPM, NPX, YARN)

3. Understand Node.js, download and install it

4. Create your first React project using the create-react-app command

5. Understand what files make up a React web application

6. Understand and run a unit test using the Jest test runner

Checkpoint!

How are you feeling?

RED - I have no idea what you’re talking about.

YELLOW - I have some questions but feel like I understand some things.

GREEN - I feel comfortable with everything you’ve said.

Homework: Test React App

1. Review the contents of the **create-react-app** project you created

2. Add the following code snippet to the test file **App.test.js**:

```
it('renders without crashing', () => {
  const div = document.createElement('div');
  ReactDOM.render(<App />,div);
});
```

3. Run test by typing **> npm test** into a terminal/command line. What are the results of the test?

4. Add the line **`import ReactDOM from 'react-dom'`** to the top of **App.test.js** and run the test again. What are the results?

5. Visit the page https://create-react-app.dev/docs/running-tests for more information about **Jest** and running tests